Dawud Gordon,
Michael Beigl,
Masayuki Iwai

A Study on the Use of Wireless Sensor Networks in a Retail Store

Braunschweig : Institute for Operating Systems and Computer Networks, 2009

Veröffentlicht: 04.06.2009

http://www.digibib.tu-bs.de/?docid=00027927

A Study on the Use of Wireless Sensor Networks in a Retail Store

Dawud Gordon, Michael Beigl
Distributed and Ubiquitous Computing
Institute for Operating Systems and Computer Networks
Technical University of Braunschweig
Germany
Email: {gordon, beigl}@ibr.cs.tu-bs.de

Masayuki Iwai
Department of Informatics and Electronics
Institute of Industrial Science
University of Tokyo
Japan
Email: masa@osoite.jp

***Abstract*—This paper presents the outcome of a formative study - experiences and requirements - on using wireless sensor network technology in retail stores. The study was carried out in several steps.**

- **First, a simple approach to implementing a wireless sensor network system was used by a developer for building a retail store application. Here we collected experiences about the problems in the application implementation process.**
- **Second, the system was deployed in a real retail store. Here we collected feedback from customers and from retail store staff about the system, its usefulness and integration into the overall daily business workflow.**
- **Third, we analyzed requirements for the next generation of improved wireless sensor network systems by performing a requirements study.**

We will report our initial results of the requirements analysis study. The findings indicate a lack of abstraction from the technical details of the system needed to enable a high-level developer to create an application for a retail scenario, and we will list some of the important issues that need to be addressed.

I. Introduction and Related Work

Today, wireless sensor networks are more difficult to implement than PC or Web-based applications. There are three reasons for this that we found. First, the connection of computing to real world workflows makes such applications complex in the sense that current software development focuses on digital workflows. Second there is the lack of software support for distributed wireless sensor network systems. Third, programming and managing wireless sensor nodes are difficult and complex tasks. Some initial solution ideas have been published for these problems. Complex integration of multiple wireless sensor networks has been addressed by Wielens et al [2], which focuses on the networking aspect. Another proposal named FLOW [3] focuses on the abstract software generation aspect. Both proposals have in common that they expect certain technical properties from their sensor nodes and thus show example implementations for one type of sensor network only. Also, both systems are complex in themselves, requiring a developer to learn a complex technical software system. Attempts to lower complexity in sensor networks by providing more abstract approaches have been developed in the context of TinyOS. In [4] for example, abstraction patterns and interfaces were developed for the design of sensor node software, but this approach only focuses on one sensor node.

We believe that part of the problem of application implementation in sensor networks is that sensor nodes are seen as complex computing devices. Instead of using the simplicity of sensor nodes to simplify development processes, we see a trend towards adding complexity to wireless sensor systems using tools, frameworks and abstraction layers. The uPart system [1] as seen in figure 2 was designed to abstract the development process of wireless sensor network applications from the technical details of the system by simplifying development. The goal was to provide developers without engineering backgrounds with a platform for development of applications without requiring knowledge of the technical details. Several applications have been built atop this platform [1][5][6]. In this paper, we focus on an application which was built for retail stores. Masayuki Iwai from the University of Tokyo designed a client behavior analysis application for a store in the Akihabara district of Tokyo, Japan, an area containing a multitude of consumer electronics vendors. In the following sections we will shortly explain the setup and the requirements that we gathered.

II. Setup

The application was deployed in one of Akihabara's many electronic store-fronts. The store was outfitted with a demonstration installation consisting of a counter-top with selected cellular phones laid out on display as shown in figure 1. Behind the counter top was a LCD display, showing an image of each cell phone model on display as well as information about that model. When a user approached the display and picked up one of the phones (a sign of interest in that model), the sensor nodes detected fluctuations in light and vibration levels. The image of that phone was moved to the center of the display and the information about that specific model was shown, as well as the popularity ranking of that model compared to the other models on display.

The system functioned by analyzing the data generated by sensor nodes attached to each cell phone which reported on the status of that phone. Each node reported how often each phone was picked up and how long the passer-by held that

Fig. 1. A Display with Items and Monitor from the Akihabara Experiment

phone in their hand before placing it back on the display. This data transmitted by the nodes was gathered by a network bridge between the sensor network and the in-house network of the shop, which transmitted that data to a central unit to be analyzed for the ranking system. The resulting rank of a specific object was created measuring user interest in the phone by analyzing the output of the vibration and light sensors on the node and thereby deducing the amount of time that users interacted with that phone as compared to others on display.

III. The Requirements Analysis Process

This situation is somewhat special because the client is a developer with specific knowledge of all stakeholders in the scenario. Although he or she is not the end user of the final product, he has direct contact with the end-users, both the end-client (the shop-keepers who would purchase such a system in order to analyze the popularity of their products) and the end-users (passers-by, who interact with the system). What the end-clients require was discovered from direct interaction with them on the part of the developer, and the end-client's wishes were obtained using a questionnaire after the fact.

In order to carry out the requirements analysis process in a scientific manner, it is necessary to select and adhere to a specific methodology. We selected the Volere [7] method because it is a free framework for requirements analysis which provides so-called ”snow-cards” for individual requirements as well as the outline of a document, which, along with the snow-cards, represents a requirements specification at the end of the process. Although useful, it is not necessary to use this or any other commercial solution.

IV. Requirements

The requirements gathered in the first round of communication with the client were very technical in nature. The fact that the client was, in this case, a developer introduces a few new variables into the development process which will be discussed here as a side-note. Under usual circumstances, requirements analysis assumes that the user has a goal and the developer can decide how to best achieve this goal in the new system. In this case however, the developer-client also operates as a developer and therefore thinks somewhat technically when it comes to his desires for the platform. What he or she may lack, however, is the knowledge of surrounding situations, such as other clients or stakeholders also involved in the process, or other effects which could create the need for compromise within the system. This incongruence introduces an unnecessary rigidity into the specification process, which could hinder the developer when optimizing the system for multiple scenarios. One possible solution is for the requirements gatherer to recognize this situation and either attempt to extract the goal of the client's technical requirements alone, which may be quite risky as an error here could be extremely expensive, or reopen discussion with the client on this requirement. This step is crucial for the system engineer to create a system which can satisfy multiple scenarios and multiple clients at the same time, by allowing the developer flexibility in his or her design.

The configuration of the individual nodes is an important part of the application. As was demonstrated by the study, this must be dynamically executable during system run-time. This requires over-the-air-configuration of the sensor nodes, which must be controllable by an application within the store, as well as be initiated from a server on the WWW. This allows the network to be able to adapt to dynamically changing conditions on the store floor, such as customer volume, or the latest market analysis, and the store-owner himself does not want to have to be aware of these adaptations. This means that the developer must account for the reconfigurations in his application, and without some method for reconfiguring from the WWW, he or she will have to execute these reconfigurations on location. This would greatly increase maintenance costs for all types of retail applications.

The system must also provide a configuration interface to reconfigure large amounts of nodes simultaneously. This is largely an ease-of-use issue for developers who often work with large quantities of nodes during development and testing stages, and need to reconfigure the entire application many times over. These new methods of node configuration must give immediate feedback to the administrator, signaling either success or failure of the configuration actions. This is necessary for the administrator to be able to weed out nodes which are still operating under old configurations in the retail application.

The system must also give instant feedback to the user as well as the administrator. The system must instantly register and display the actions of a user who is interacting with the system. If this does not occur, or if the delay is too long, the user does not have the feeling of interaction with the system. A good example of this is that when a user picks up a cell phone on display, the display monitor must instantly reflect this action.

The sensory attributes of the sensor nodes are also important for the development platform. It is vital to the scenario that the vibration sensor be accurate and sensitive enough to be able to recognize human interaction with the sensor nodes, in this case if an object to which it is attached is being picked up or held in the hand of the user. Also the light sensor must be able to operate in indoor and outdoor environments, as the retail environment extends from the retail store interior, where there

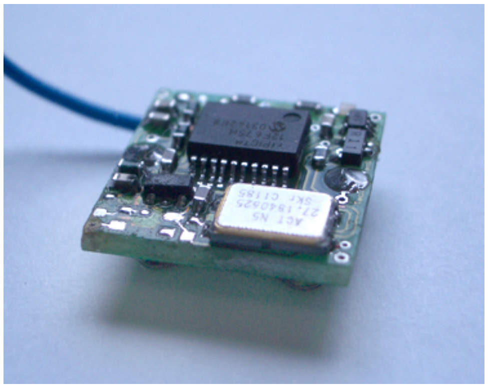

Fig. 2. The uPart Wireless Sensor Node Used in the Akihabara Study

are low light conditions, to the store front in the street where there may be very high light levels including direct sunlight. Because of this it is important that the light sensor neither saturate under normal day-light conditions, nor bottom out in low-light environments.

Most retail environments have opening and closing hours and in order to extend battery-powered network lifetimes and reduce maintenance costs it is important to take advantage of these phases. The system should provide a method for going into a dormant mode after closing time in order to reduce power consumption of the sensor nodes. In the same grain, the system should also support a wake function based on these hours, as well as a manual wake-up override. In order to keep the systems as low-maintenance as possible for the end-clients, they must be able to see the remaining energy level as well as the time remaining in current modus of the nodes' batteries. Based on this information, the shop-keeper should be notified 24 hours before a node will finally run out of energy. Also, the system should be able to provide some sort of node locating assistance. This will help the shop keeper carry out normal maintenance processes, such as battery changes or node relocations.

The scenario of an Akihabara store-front is actually two separate scenarios with a large amount of overlap. The first scenario is the one previously discussed, for which a range of at least 10 meters is necessary for the nodes. The other scenario is that sensor nodes are attached to various objects of the shop-keeper's choosing throughout the store which are then monitored by an application accessible only to the shop-keeper. This allows him to monitor the popularity of his products among his clients with or without their knowledge. For this scenario, a range of 40 meters is necessary in order to cover the floor area of the majority of the stores in the Akihabara district. The distinction between these two scenarios is important for retail scenarios as it defines a double network range requirement: one single network for an entire store, or multiple applications/networks per retail store environment.

Continuing in the direction of multiple applications per environment, it is necessary to be able to group certain nodes together in order to create separate applications which operate in the same environment. Also, not only intra-retail-store interference may occur, but inter-store collisions as well between two separate retail stores operating the same product. For example, if the resulting product of this trial would become popular in the Akihabara district, it would rapidly become necessary to implement some sort of interference protection for the case where two storefronts implement the same application with a wireless network reception overlap. The resulting mechanism must create groups of nodes which can be configured together, and also provides a measure of interference protection between two applications running side-by-side.

V. Conclusion

The retail store application acted quite well as a method for gathering experience on implementing a wireless sensor network in a retail environment. Our overall impression is that there are still some areas of wireless sensor networks which lack a certain level of abstraction from the underlying technical principles. We have also presented a list of requirements which are suggestions for improving these systems to allow high-level developers to design their own systems. We learned that users need to receive immediate feedback from the system in order to validate their interaction. Also, store owners want to have as little as possible to do with the technical side of the network, and for this reason an acknowledged over-the-air-configuration method for one or multiple nodes from points inside and outside of the store's network must be made available. The nodes themselves must incorporate sensors and range capabilities which can support the varying conditions in a retail store. The nodes must also be able to adapt to the working hours of such a store and be able to withstand possible interference from neighboring wireless sensor network applications. We are already working on the next generation of sensor nodes, which with the help of these requirements, will be one step closer to enabling high-level developers to create their own wireless sensor network solutions.

References

[1] M. Beigl, A. Krohn, T. Riedel, T. Zimmer, C. Decker, M. Isomura *The uPart Experience: Building a Wireless Sensor Network*, IEEE/ACM Conference on Information Processing in Sensor Networks (IPSN), 2006

[2] S. Wielens, M. Galetzka and P. Schneider, *Design support for Wireless Sensor Networks Based on the IEEE 802.15.4 Standard*, Personal, Indoor and Mobile Radio Communications (PIMRC), 15-18 Sept. 2008

[3] T. Naumowicz. *Managing Data Flows with Flow: A Software Factory for Wireless Sensor Networks.*, Poster. Microsoft Academic Days 2008, Berlin, Germany, November 2008

[4] D. Gay, P. Levis, and D. Culler, *Software Design Patterns for TinyOS.* In Proceedings of the 2005 ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools For Embedded Systems (LCTES) Chicago, Illinois, June 2005

[5] J. Nakazawa and H. Tokuda, *A Middleware Framework for Sharing Sensor Nodes among Smart Spaces.* Fourth International Conference on Networked Sensing Systems, 2007. (INSS'07)

[6] M. Isomura, T. Riedel, C. Decker, M. Beigl, H. Horiuchi, *Sharing sensor networks, Sixth International Workshop on Smart Appliances and Wearable Computing*, International Workshop on Smart Appliances and Wearable Computing (IWSAWC) 2006, Lisbon, Portugal: Proceedings of the ICDCS 2006, IEEE Computer Society

[7] S. and J. Robertson, *Mastering the Requirements Process*, Addison-Wesley Professional, 1999, 2nd Edition